CLAIMS

What is claimed is:

1. A method of designing a plurality of integrated circuits (ICs), said method comprising:

partitioning a technology independent RTL (register transfer level) netlist between said plurality of ICs.

2. A method as in claim 1 further comprising:

compiling a hardware description language (HDL) code, wherein said technology independent RTL netlist is produced after compiling said HDL code.

3. A method as in claim 2 wherein said ICs each comprise a programmable logic device and wherein said partitioning comprises assigning a portion of said technology independent RTL netlist to one of said plurality of ICs.

4. A method as in claim 2 further comprising:

mapping said technology independent RTL netlist to a selected technology architecture.

5. A method as in claim 4 wherein said mapping is performed after said partitioning.

6. A method as in claim 4 further comprising:

performing a place and route operation after said mapping to implement said ICs in said selected technology architecture.

7. A method as in claim 4 further comprising:

optimizing a design of each of said ICs after said partitioning.

8. A method as in claim 7 wherein said optimizing optimizes each of said ICs by removing duplicative logic or input/outputs.

9. A method as in claim 4 wherein said HDL code is created without regard to said partitioning.

10. A method as in claim 7 wherein said optimizing and said mapping are performed after said partitioning.

11. A method as in claim 4 further comprising:

mapping portions of said technology independent RTL netlist to a selected technology architecture wherein estimates of IC resources are obtained from said mapping portions and wherein said mapping portions is performed after said compiling and before said mapping.

12. A method as in claim 4 further comprising:

optimizing interconnects between modules of said technology independent RTL netlist before said partitioning.

13. A method as in claim 11 wherein said estimates are used to decide how to perform said partitioning.

14. A method as in claim 13 wherein a user considers said estimates and selects a command to decide how to perform said partitioning.

15. A method as in claim 4 wherein said ICs each comprise a programmable logic device and wherein said method further comprises:

testing a prototype of a system with said ICs;

performing a synthesis of said HDL code to generate at least one Application Specific Integrated Circuit (ASIC).

16. A machine readable medium containing a plurality of executable instructions, which when executed on a digital processing system cause said digital processing system to perform a method of designing a plurality of integrated circuits (ICs), said method comprising:

partitioning a technology independent RTL (register transfer level) netlist between said plurality of ICs.

17. A machine readable medium as in claim 16, wherein said method further comprises:

compiling a hardware description language (HDL) code, wherein said technology independent RTL netlist is produced after compiling said HDL code.

18. A machine readable medium as in claim 17 wherein said ICs each comprise a programmable logic device.

19. A machine readable medium as in claim 17, wherein said method further comprises:

mapping said technology independent RTL netlist to a selected technology architecture.

20. A machine readable medium as in claim 19 wherein said mapping is performed after said partitioning.

21. A machine readable medium as in claim 19, wherein said method further comprises:

performing a place and route operation after said mapping to implement said ICs in said selected technology architecture.

22. A machine readable medium as in claim 19, wherein said method further comprises:

optimizing a design of each of said ICs after said partitioning.

23. A machine readable medium as in claim 22 wherein said optimizing optimizes each of said ICs by removing duplicative logic or input/outputs.

24. A machine readable medium as in claim 19 wherein said HDL code is created without regard to said partitioning.

25. A machine readable medium as in claim 22 wherein said optimizing and said mapping are performed after said partitioning.

26. A machine readable medium as in claim 19, wherein said method further comprises:

mapping portions of said technology independent RTL netlist to a selected technology architecture wherein estimates of IC resources are obtained from said mapping portions and wherein said mapping portions is performed after said compiling and before said mapping.

27. A machine readable medium as in claim 19, wherein said method further comprises:

optimizing interconnects between modules of said technology independent RTL netlist before said partitioning.

28. A machine readable medium as in claim 26 wherein said estimates are used to decide how to perform said partitioning.

29. A machine readable medium as in claim 28 wherein a user considers said estimates and selects a command to decide how to perform said partitioning.

30. A machine readable medium as in claim 19 wherein said ICs each comprise a programmable logic device and wherein said method further comprises:

testing a prototype of a system with said ICs;

performing a synthesis of said HDL code to generate at least one Application Specific Integrated Circuit (ASIC).

31. A system of designing a plurality of integrated circuits (ICs), said system comprising:

means for displaying a representation of said plurality of ICs;

means for partitioning a technology independent RTL (register transfer level) netlist between said plurality of ICs.

32. A system as in claim 31 further comprising:

means for compiling a hardware description language (HDL) code, wherein said technology independent RTL netlist is produced after compiling said HDL code.

33. A system as in claim 32 wherein said ICs each comprise a programmable logic device.

34. A system as in claim 32 further comprising:

means for mapping said technology independent RTL netlist to a selected technology architecture.

35. A system as in claim 34 wherein said mapping is performed after said partitioning.

36. A system as in claim 34 further comprising:

means for performing a place and route operation after said mapping to implement said ICs in said selected technology architecture.

37. A system as in claim 34 further comprising:

means for optimizing a design of each of said ICs after said partitioning.

38. A system as in claim 37 wherein said optimizing optimizes each of said ICs by removing duplicative logic or input/outputs.

39. A system as in claim 34 wherein said HDL code is created without regard to said partitioning.

40. A system as in claim 37 wherein said optimizing and said mapping are performed after said partitioning.

41. A system as in claim 34 further comprising:

means for mapping portions of said technology independent RTL netlist to a selected technology architecture wherein estimates of IC resources are obtained from said mapping portions and wherein said mapping portions is performed after said compiling and before said mapping.

42. A system as in claim 34 further comprising:

means for optimizing interconnects between modules of said technology independent RTL netlist before said partitioning.

43. A system as in claim 41 wherein said estimates are used to decide how to perform said partitioning.

44. A system as in claim 43 wherein a user considers said estimates and selects a command to decide how to perform said partitioning.

45. A system as in claim 34 wherein said ICs each comprise a programmable logic device and wherein said system further comprises:

means for testing a prototype of a system with said ICs;

means for performing a synthesis of said HDL code to generate at least one Application Specific Integrated Circuit (ASIC).

46. A digital processing system for use in designing a plurality of integrated circuits (ICs), said digital processing system comprising:

a display device;

a memory;

a processor coupled to said memory and to said display device, said processor partitioning a technology independent RTL (register transfer level) netlist between representations of said plurality of ICs, said technology independent RTL netlist being stored in said memory.

47. A digital processing system as in claim 46 wherein said processor compiles a hardware description language (HDL) code to produce said technology independent RTL netlist.

48. A digital processing system as in claim 47 wherein said ICs each comprise a programmable logic device.

49. A digital processing system as in claim 47 wherein said processor maps said technology independent RTL netlist to a selected technology architecture.

50. A digital processing system as in claim 49 wherein said processor maps said technology independent RTL netlist after said processor partitions said technology independent RTL netlist.

51. A digital processing system as in claim 49 wherein said processor performs a place and route operation after said processor maps said technology independent RTL netlist, wherein said place and route operation creates a representation of circuitry in said selected technology architecture.

52. A digital processing system as in claim 50 wherein said processor optimizes a design of each of said ICs after said processor partitions said technology independent RTL netlist.

53. A digital processing system as in claim 51 wherein said processor maps said technology independent RTL netlist after said processor partitions said technology independent RTL netlist.

54. A digital processing system as in claim 53 wherein said processor maps portions of said technology independent RTL netlist to said selected technology

architecture to generate estimates of IC resources and wherein said processor maps said portions after said processor compiles said HDL code.

55. A digital processing system as in claim 53 wherein said processor displays said estimates on said display device and stores said estimates in said memory.

56. A digital processing system as in claim 55 wherein said processor displays graphical representations of said plurality of ICs on said display device and displays on said display device representations of portions of said technology independent RTL netlist and wherein said processor performs said partitioning in response to a command from a user.

57. A method of designing a plurality of integrated circuits (ICs), said method comprising:

compiling a hardware description language (HDL) code to produce an RTL netlist representation which specifies said plurality of ICs;

selecting logic designed for placement on one of said plurality of ICs and replicating said logic for placement on another one of said plurality of ICs.

58. A method as in claim 57 further comprising:

partitioning said RTL netlist representation to specify said plurality of ICs.

59. A method as in claim 58 further comprising:

mapping said RTL netlist representation to a selected technology architecture, wherein said RTL netlist representation is technology independent.

60. A method as in claim 59 wherein each of said plurality of ICs comprises a programmable logic device.

61. A method as in claim 59 wherein said mapping is performed after said partitioning.

62. A method as in claim 61 further comprising:

performing a place and route operation after said mapping to implement logic in said plurality of ICs.

63. A method as in claim 62 further comprising:

optimizing a design of each of said plurality of ICs after said partitioning.

64. A method as in claim 57 wherein said HDL code is created without specifying a plurality of ICs.

65. A method as in claim 59 further comprising:

mapping portions of said RTL netlist representation to said selected technology architecture wherein estimates of IC resources are obtained from said

mapping portions and wherein said mapping portions is performed after said compiling and before said mapping.

66. A method as in claim 65 wherein said estimates are used to decide how to perform said selecting and replicating.

67. A method as in claim 60 further comprising:

testing a prototype of a system with said plurality of ICs;

performing, after said testing, a synthesis on said HDL code to generate at least one Application Specific Integrated Circuit (ASIC).

68. A machine readable medium containing a plurality of executable instructions, which when executed on a digital processing system cause said digital processing system to perform a method of designing a plurality of integrated circuits (ICs), said method comprising:

compiling a hardware description language (HDL) code to produce an RTL netlist representation which specifies said plurality of ICs;

selecting logic designed for placement on one of said plurality of ICs and replicating said logic for placement on another one of said plurality of ICs.

69. A machine readable medium as in claim 68, said method further comprising: partitioning said RTL netlist representation to specify said plurality of ICs.

70. A machine readable medium as in claim 69, said method further comprising:
mapping said RTL netlist representation to a selected technology architecture,
wherein said RTL netlist representation is technology independent.

71. A machine readable medium as in claim 70, wherein each of said plurality of ICs comprises a programmable logic device.

72. A machine readable medium as in claim 70, wherein said mapping is performed after said partitioning.

73. A machine readable medium as in claim 72, said method further comprising:
performing a place and route operation after said mapping to implement logic in said plurality of ICs.

74. A machine readable medium as in claim 73, said method further comprising:
optimizing a design of each of said plurality of ICs after said partitioning.

75. A machine readable medium as in claim 68 wherein said HDL code is created without specifying a plurality of ICs.

76. A machine readable medium as in claim 70, said method further comprising:

mapping portions of said RTL netlist representation to said selected technology architecture wherein estimates of IC resources are obtained from said mapping portions and wherein said mapping portions is performed after said compiling and before said mapping.

77. A machine readable medium as in claim 76 wherein said estimates are used to decide how to perform said selecting and replicating.

78. A machine readable medium as in claim 71, said method further comprising:

testing a prototype of a system with said plurality of ICs;

performing, after said testing, a synthesis on said HDL code to generate at least one Application Specific Integrated Circuit (ASIC).

79. A system for designing a plurality of integrated circuits (ICs), said system comprising:

means for compiling a hardware description language (HDL) code to produce an RTL netlist representation which specifies said plurality of ICs;

means for selecting logic designed for placement on one of said plurality of ICs and replicating said logic for placement on another one of said plurality of ICs.

80. A system as in claim 79 further comprising:

means for partitioning said RTL netlist representation to specify said plurality of ICs.

81. A system as in claim 80 further comprising:

means for mapping said RTL netlist representation to a selected technology architecture, wherein said RTL netlist representation is technology independent.

82. A system as in claim 81 wherein each of said plurality of ICs comprises a programmable logic device.

83. A system as in claim 81 wherein said mapping is performed after said partitioning.

84. A system as in claim 83 further comprising:

means for performing a place and route operation after said mapping to implement logic in said plurality of ICs.

85. A system as in claim 84 further comprising:

means for optimizing a design of each of said plurality of ICs after said partitioning.

86. A system as in claim 79 wherein said HDL code is created without specifying a plurality of ICs.

87. A system as in claim 81 further comprising:

means for mapping portions of said RTL netlist representation to said selected technology architecture wherein estimates of IC resources are obtained from said mapping portions and wherein said mapping portions is performed after said compiling and before said mapping.

88. A system as in claim 87 wherein said estimates are used to decide how to perform said selecting and replicating.

89. A system as in claim 82 further comprising:

means for testing a prototype of a system with said plurality of ICs;

means for performing, after said testing, a synthesis on said HDL code to generate at least one Application Specific Integrated Circuit (ASIC).

90. A method of designing a plurality of integrated circuits (ICs), said method comprising:

compiling a hardware description language (HDL) code to produce an RTL netlist representation;

selecting one RTL component in said RTL netlist representation and splitting said one RTL component into a first RTL component designed for

placement on a first IC and a second RTL component designed for placement on a second IC.

91. A method as in claim 90 further comprising:

partitioning said RTL netlist representation to specify said first IC and said second IC.

92. A method as in claim 91 further comprising:

mapping said RTL netlist representation to a selected technology architecture, wherein said RTL netlist representation is technology independent.

93. A method as in claim 92 wherein each of said first IC and said second IC comprises a programmable logic device.

94. A method as in claim 92 wherein said mapping is performed after said partitioning.

95. A method as in claim 94 further comprising:

performing a place and route operation after said mapping to implement logic for said first RTL component for placement on said first IC and to implement logic for said second RTL component for placement on said second IC.

96. A method as in claim 92 wherein said one RTL component is created as a unitary object after said compiling.

97. A method as in claim 92 wherein said one RTL component is a large RTL component having more inputs/outputs than is available on said first IC or said second IC.

98. A method as in claim 92 further comprising:

mapping portions of said RTL netlist representation to said selected technology architecture wherein estimates of IC resources are obtained from said mapping portions and wherein said mapping portions is performed after said compiling and before said mapping.

99. A method as in claim 98 wherein said selecting and said splitting are performed after said compiling and before said mapping.

100. A method as in claim 99 wherein said estimates are used to determine how to split said one RTL component.

101. A machine readable medium containing a plurality of executable instructions, which when executed on a digital processing system cause said digital processing system to perform a method of designing a plurality of integrated circuits (ICs), said method comprising:

compiling a hardware description language (HDL) code to produce an RTL netlist representation;

selecting one RTL component in said RTL netlist representation and splitting said one RTL component into a first RTL component designed for placement on a first IC and a second RTL component designed for placement on a second IC.

102. A machine readable medium as in claim 101, wherein said method further comprises:

partitioning said RTL netlist representation to specify said first IC and said second IC.

103. A machine readable medium as in claim 102, wherein said method further comprises:

mapping said RTL netlist representation to a selected technology architecture, wherein said RTL netlist representation is technology independent.

104. A machine readable medium as in claim 103 wherein each of said first IC and said second IC comprises a programmable logic device.

105. A machine readable medium as in claim 103 wherein said mapping is performed after said partitioning.

106. A machine readable medium as in claim 105, wherein said method further comprises:

performing a place and route operation after said mapping to implement logic for said first RTL component for placement on said first IC and to implement logic for said second RTL component for placement on said second IC.

107. A machine readable medium as in claim 103 wherein said one RTL component is created as a unitary object after said compiling.

108. A machine readable medium as in claim 103 wherein said one RTL component is a large RTL component having more inputs/outputs than is available on said first IC or said second IC.

109. A machine readable medium as in claim 103, wherein said method further comprises:

mapping portions of said RTL netlist representation to said selected technology architecture wherein estimates of IC resources are obtained from said mapping portions and wherein said mapping portions is performed after said compiling and before said mapping.

110. A machine readable medium as in claim 109 wherein said selecting and said splitting are performed after said compiling and before said mapping.

111. A machine readable medium as in claim 110 wherein said estimates are used to determine how to split said one RTL component.

112. A system for designing a plurality of integrated circuits (ICs), said system comprising:

means for compiling a hardware description language (HDL) code to produce an RTL netlist representation;

means for selecting one RTL component in said RTL netlist representation and splitting said one RTL component into a first RTL component designed for placement on a first IC and a second RTL component designed for placement on a second IC.

113. A system as in claim 112 further comprising:

means for partitioning said RTL netlist representation to specify said first IC and said second IC.

114. A system as in claim 113 further comprising:

means for mapping said RTL netlist representation to a selected technology architecture, wherein said RTL netlist representation is technology independent.

115. A system as in claim 114 wherein each of said first IC and said second IC comprises a programmable logic device.

116. A system as in claim 114 wherein said mapping is performed after said partitioning.

117. A system as in claim 116 further comprising:

means for performing a place and route operation after said mapping to implement logic for said first RTL component for placement on said first IC and to implement logic for said second RTL component for placement on said second IC.

118. A system as in claim 114 wherein said one RTL component is created as a unitary object after said compiling.

119. A system as in claim 114 wherein said one RTL component is a large RTL component having more inputs/outputs than is available on said first IC or said second IC.

120. A system as in claim 114 further comprising:

means for mapping portions of said RTL netlist representation to said selected technology architecture wherein estimates of IC resources are obtained

from said mapping portions and wherein said mapping portions is performed after said compiling and before said mapping.

121. A system as in claim 120 wherein said selecting and said splitting are performed after said compiling and before said mapping.

122. A system as in claim 121 wherein said estimates are used to determine how to split said one RTL component.

123. A method of designing a plurality of integrated circuits (ICs), said method comprising:

assigning a first portion of a technology independent RTL (register transfer level) netlist for placement on a first IC of said plurality of ICs;

assigning a second portion of said technology independent RTL netlist to a second IC of said plurality of ICs.

124. A method as in claim 123 further comprising:

compiling a hardware description language (HDL) code, wherein said technology independent RTL netlist is produced after compiling said HDL code.

125. A method as in claim 124 wherein said ICs each comprise a programmable logic device.

126. A method as in claim 124 further comprising:

mapping said technology independent RTL netlist to a selected technology architecture.

127. A method as in claim 126 wherein said mapping is performed after said assigning.

128. A method as in claim 126 further comprising:

performing a place and route operation after said mapping to implement said ICs in said selected technology architecture.

129. A method as in claim 126 further comprising:

optimizing a design of each of said ICs after said assigning.

130. A method as in claim 129 wherein said optimizing optimizes each of said ICs by removing duplicative logic or input/outputs.

131. A method as in claim 126 wherein said HDL code is created without regard to said assigning.

132. A method as in claim 129 wherein said optimizing and said mapping are performed after said assigning.

133. A method as in claim 126 further comprising:

mapping portions of said technology independent RTL netlist to a selected technology architecture wherein estimates of IC resources are obtained from said mapping portions and wherein said mapping portions is performed after said compiling and before said mapping.

134. A method as in claim 126 further comprising:

optimizing interconnects between modules of said technology independent RTL netlist before said assigning.

135. A method as in claim 133 wherein said estimates are used to decide how to perform said assigning.

136. A method as in claim 135 wherein a user considers said estimates and selects a command to decide how to perform said assigning.

137. A method as in claim 126 wherein said ICs each comprise a programmable logic device and wherein said method further comprises:

testing a prototype of a system with said ICs;

performing a synthesis of said HDL code to generate at least one Application Specific Integrated Circuit (ASIC).